Digital Image Processing & Machine Vision

SESSION # 6

Case Study

Transfer Learning for Image Denoising

Case Background

Image denoising is a fundamental task in image processing, aiming to remove noise from corrupted images while preserving important details and structures.

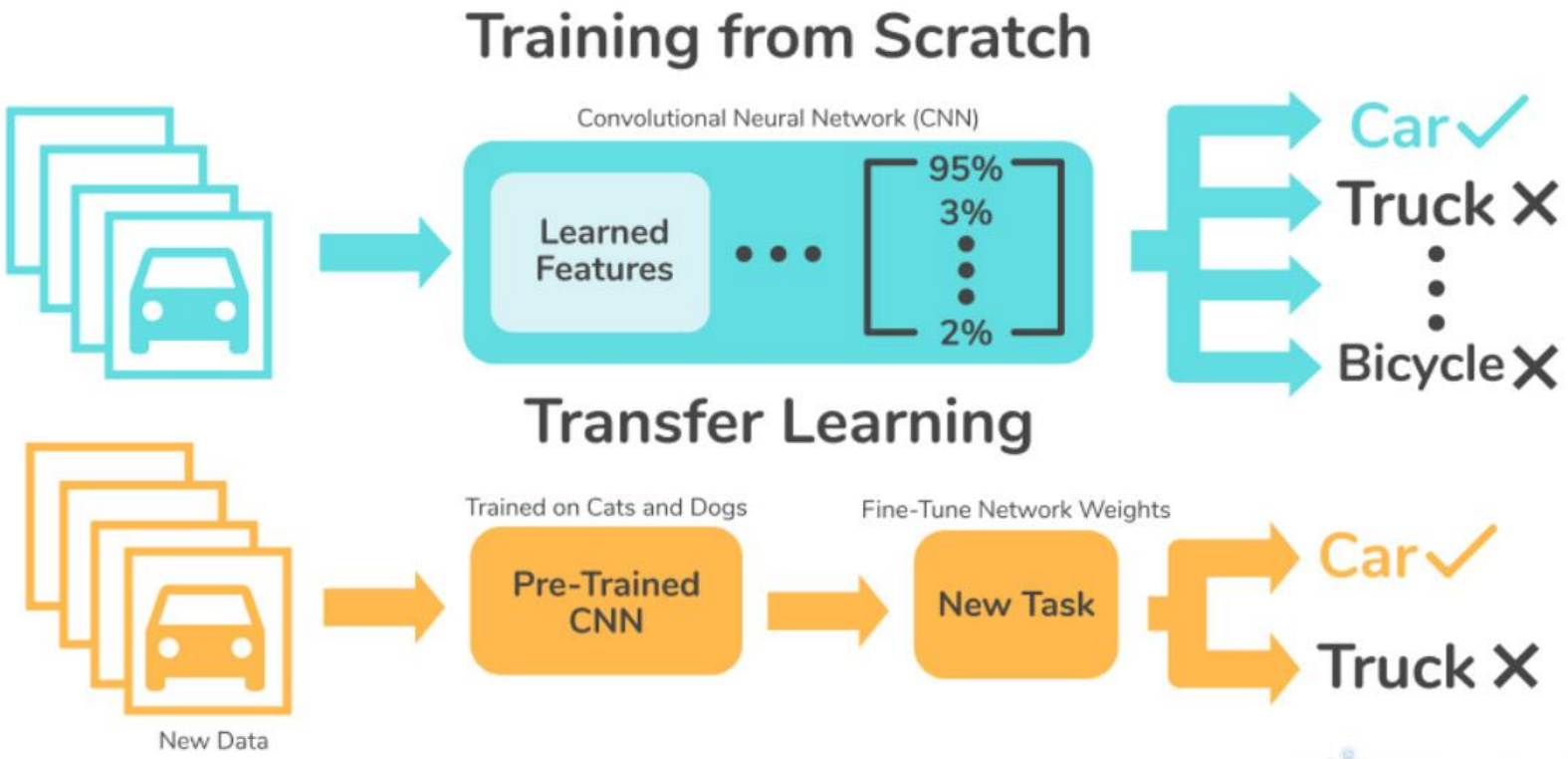
Traditional denoising methods, such as Gaussian filtering or wavelet transforms, often struggle with complex noise patterns and may blur fine details.

Deep learning-based approaches, particularly Convolutional Neural Networks (CNNs), have shown remarkable success in denoising tasks.

However, training such models from scratch requires large datasets and significant computational resources.

Transfer learning offers a solution by leveraging pre-trained models and adapting them to denoising tasks.

Transfer Learning



Analogy for Transfer Learning

- Example-1
  - Imagine you're learning to **drive a car** after already knowing how to **ride a bicycle**.

Pre-trained Model (Bicycle Skills)

- You’ve spent years riding a bicycle. You’ve learned skills like **balancing, steering, and understanding traffic rules**.
- These skills are like the **pre-trained model**—they’ve been learned from a lot of practice (the "bicycle dataset").

New Task (Driving a Car)

- Instead of starting from scratch, you **transfer** your existing knowledge of balancing, steering, and traffic rules from riding a bicycle to driving a car.
- You already know how to navigate roads, follow traffic signals, and avoid obstacles, so you don’t need to learn these basics again. This is like **reusing the pre-trained features** from your bicycle experience.

Fine-Tuning (Adapting to Driving a Car)

- However, driving a car is different from riding a bicycle. For example:
  - You need to learn how to use the **accelerator, brake, and clutch** (new skills specific to driving).
  - You also need to get used to the **size and weight of the car** and how it handles on the road.
- This is like **fine-tuning** the pre-trained model. You keep the general road skills (frozen layers) but adapt to the specifics of driving a car (train new layers).

Analogy for Transfer Learning

- Example-2
  - Imagine you're learning to play the **guitar** after already knowing how to play the **piano**.

Pre-trained Model (Piano Skills)

- You’ve spent years learning the piano. You understand musical concepts like **scales, chords, rhythm, and reading sheet music**.
- These skills are like the **pre-trained model**—they’ve been learned from a large "dataset" (your piano practice).

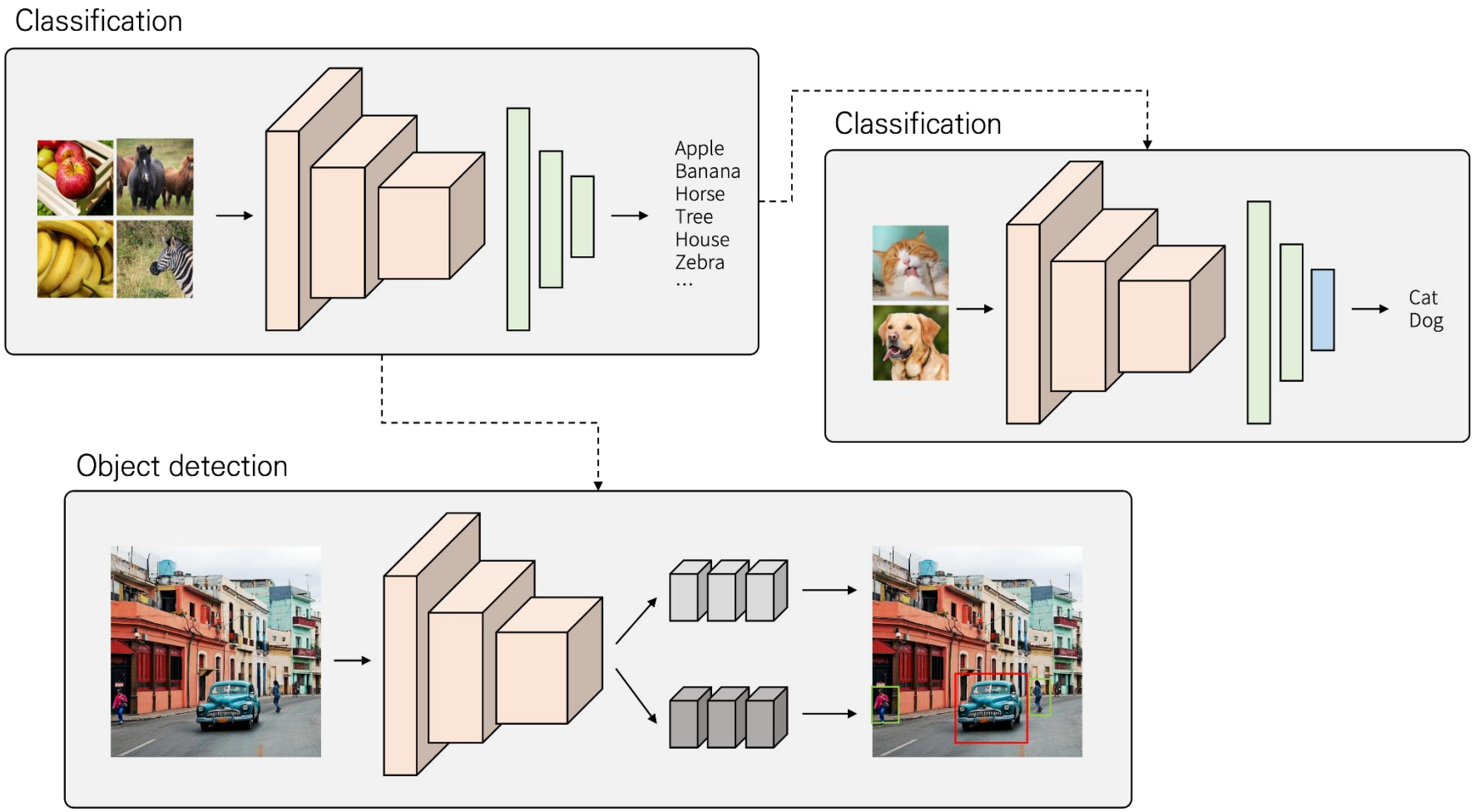
Fine-Tuning (Adapting to Guitar)

- However, the guitar is different from the piano. For example:
  - You need to learn how to **strum** and **fingerpick** (new skills specific to the guitar).
  - You also need to get used to the **fretboard** and **chord shapes**.
- This is like **fine-tuning** the pre-trained model. You keep the general musical knowledge (frozen layers) but adapt to the specifics of the guitar (train new layers).

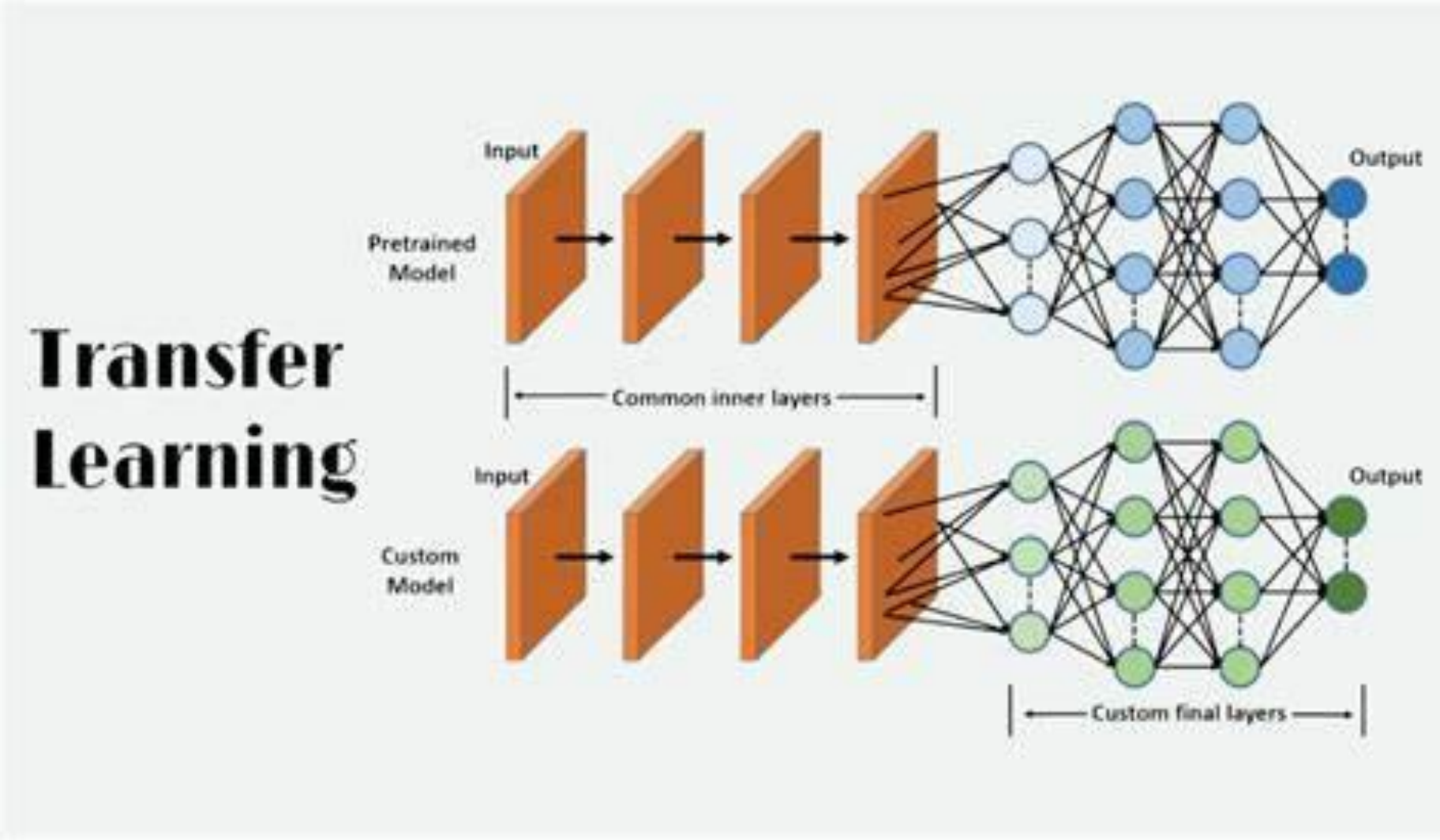
Benefits of Transfer Learning

- Because you already know music theory and rhythm from the piano, learning the guitar is **much faster** than if you were starting from scratch.
- You don’t need to spend as much time practicing the basics, and you can focus on the unique aspects of the guitar.

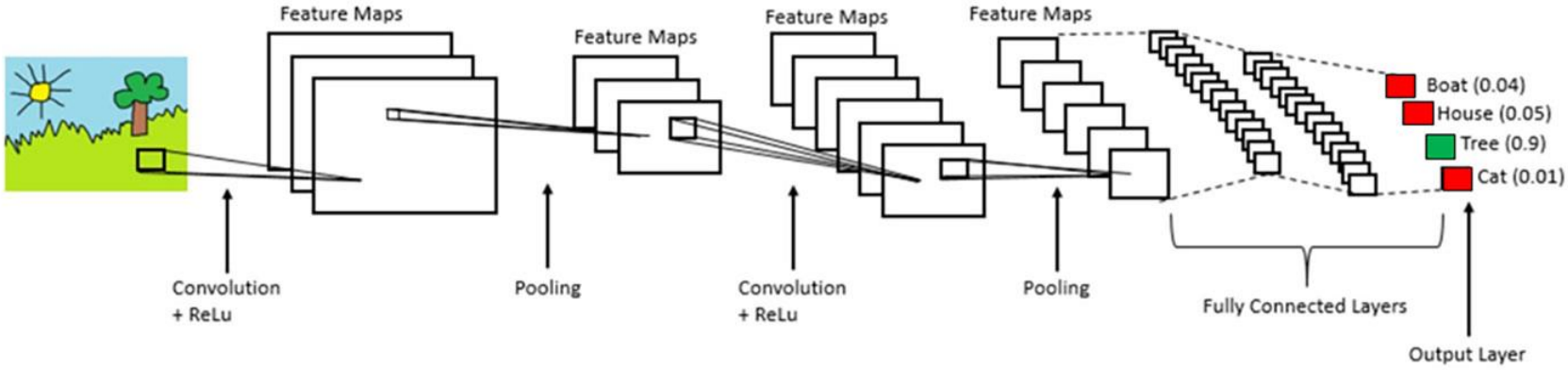
Transfer Learning



Transfer Learning



Transfer Learning



Problem Statement

- To develop an efficient image denoising system for medical imaging, where noise can obscure critical details in X-ray or MRI images.
  - Limited access to labeled medical image data but wants to leverage existing pre-trained models to achieve high-quality denoising.

Dataset Preparation & Model Adaption

Dataset Preparation

A small dataset of noisy medical images and their corresponding clean versions is collected.

The dataset is augmented with transformations (e.g., rotation, flipping) to increase its effective size.

Model Adaptation

The pre-trained model is fine-tuned on the medical image dataset. The earlier layers of the model, which capture low-level features like edges and textures, are frozen, while the later layers are retrained to adapt to the specific noise characteristics of medical images.

Alternatively, the model is modified to include additional layers specifically designed for denoising

Training & Evaluation

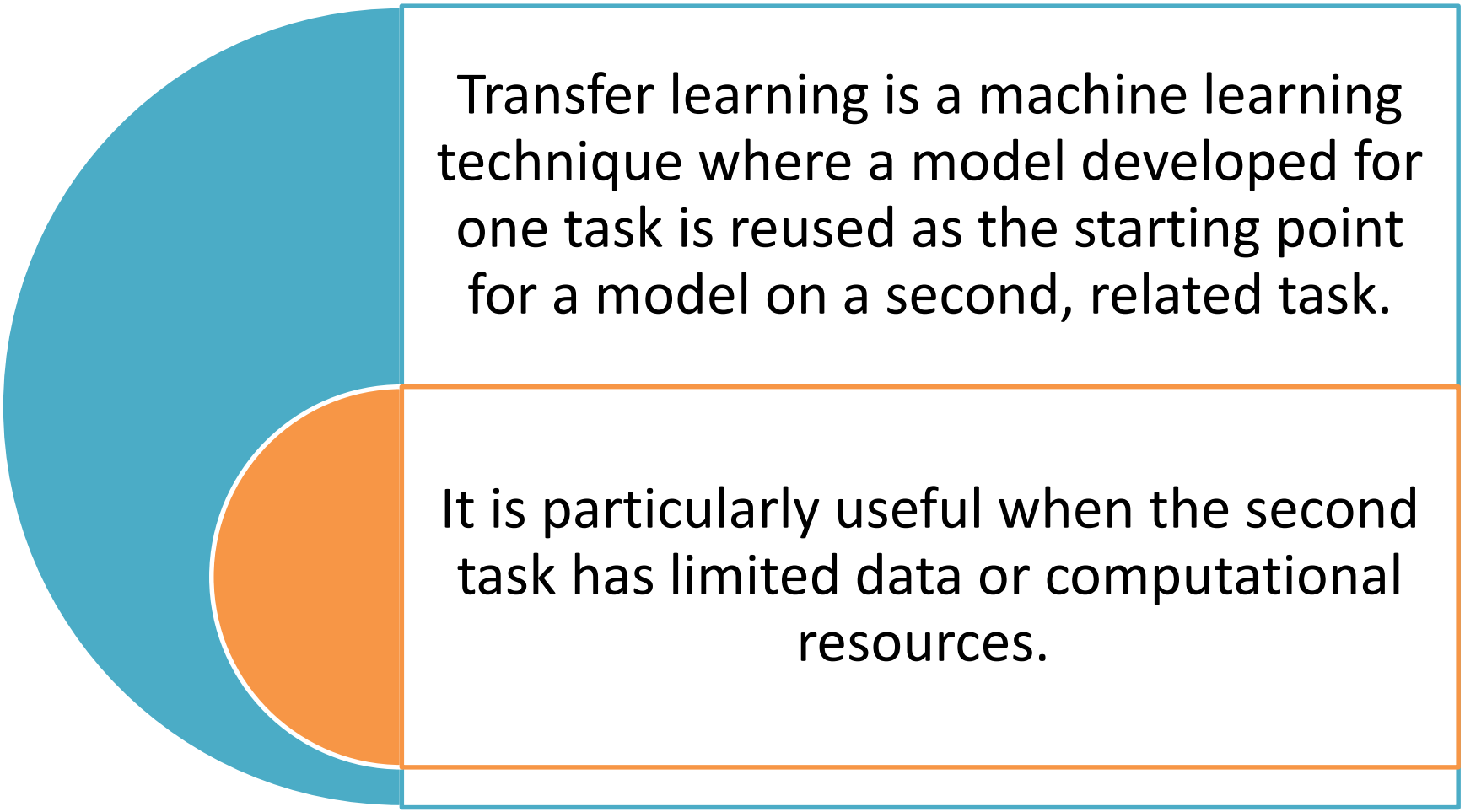
Training

- The model is trained using a loss function such as **Mean Squared Error (MSE)** or **Structural Similarity Index (SSIM)** to minimize the difference between the denoised output and the clean ground truth images.
- Transfer learning significantly reduces training time and computational cost compared to training from scratch.

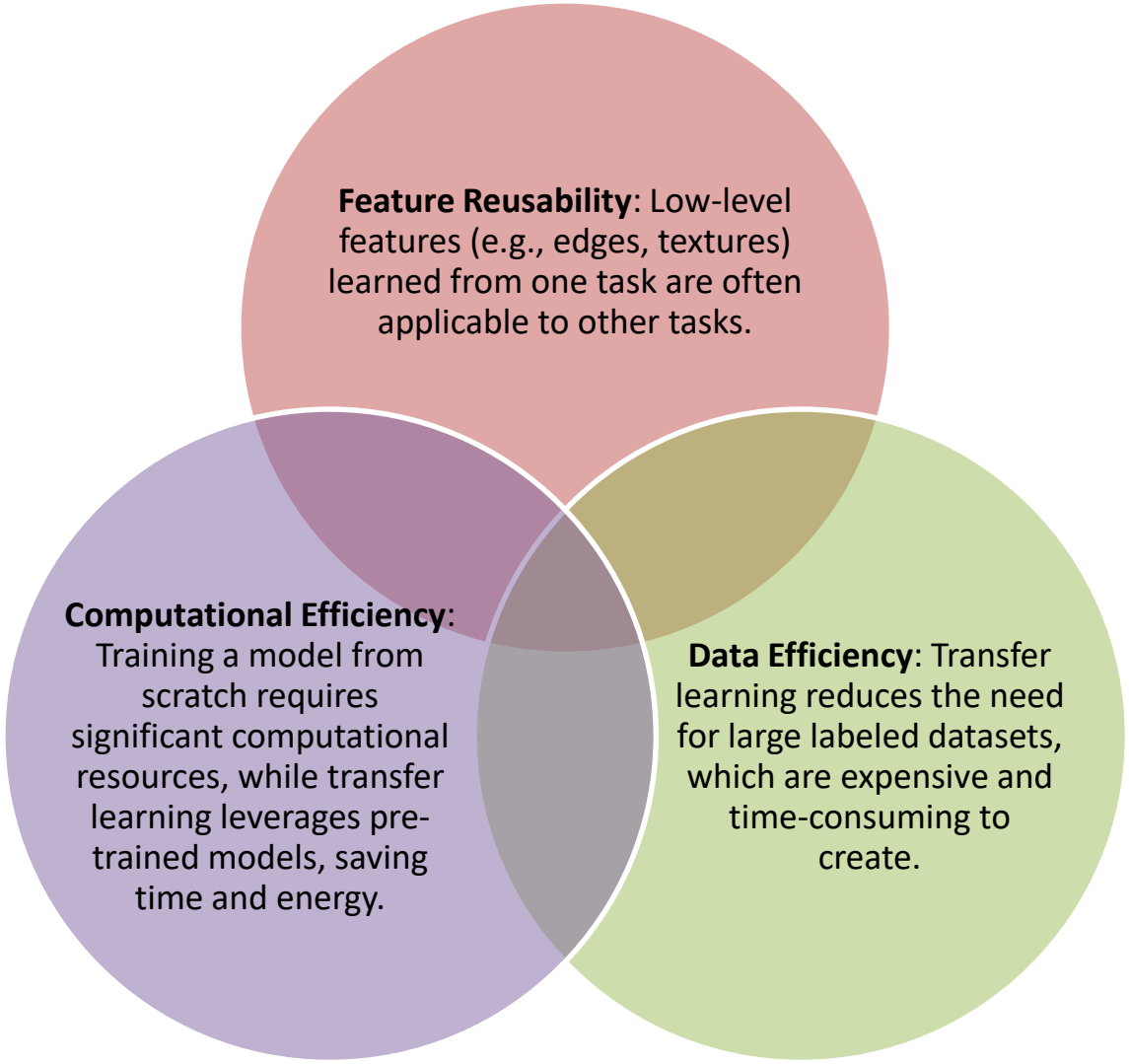
Evaluation

- The denoising performance is evaluated using metrics like **Peak Signal-to-Noise Ratio (PSNR)** and **SSIM**.
- The model is compared against traditional denoising methods and a baseline CNN trained from scratch.

Reflection of Transfer Learning


Transfer learning is a machine learning technique where a model developed for one task is reused as the starting point for a model on a second, related task.
It is particularly useful when the second task has limited data or computational resources.

Why Transfer Learning Works

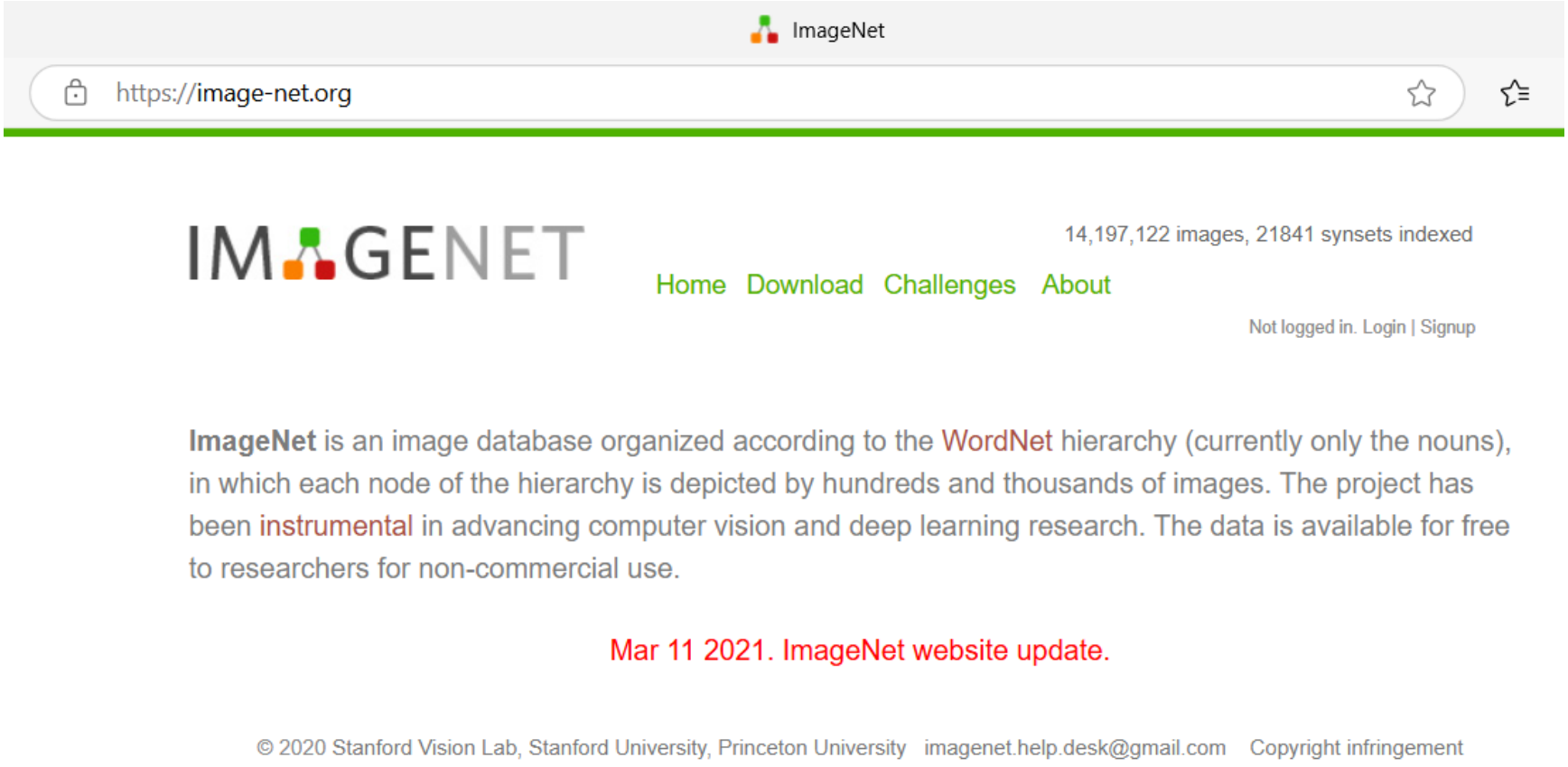

Feature Reusability: Low-level features (e.g., edges, textures) learned from one task are often applicable to other tasks.
Computational Efficiency: Training a model from scratch requires significant computational resources, while transfer learning leverages pre-trained models, saving time and energy.
Data Efficiency: Transfer learning reduces the need for large labeled datasets, which are expensive and time-consuming to create.

Pre-trained Model Selection

- The team selects a pre-trained CNN model, such as **ResNet** or **DnCNN** (Denoising Convolutional Neural Network), which has been trained on a large dataset like ImageNet or a generic denoising dataset.
  - These models have learned useful feature representations that can be adapted to the denoising task.

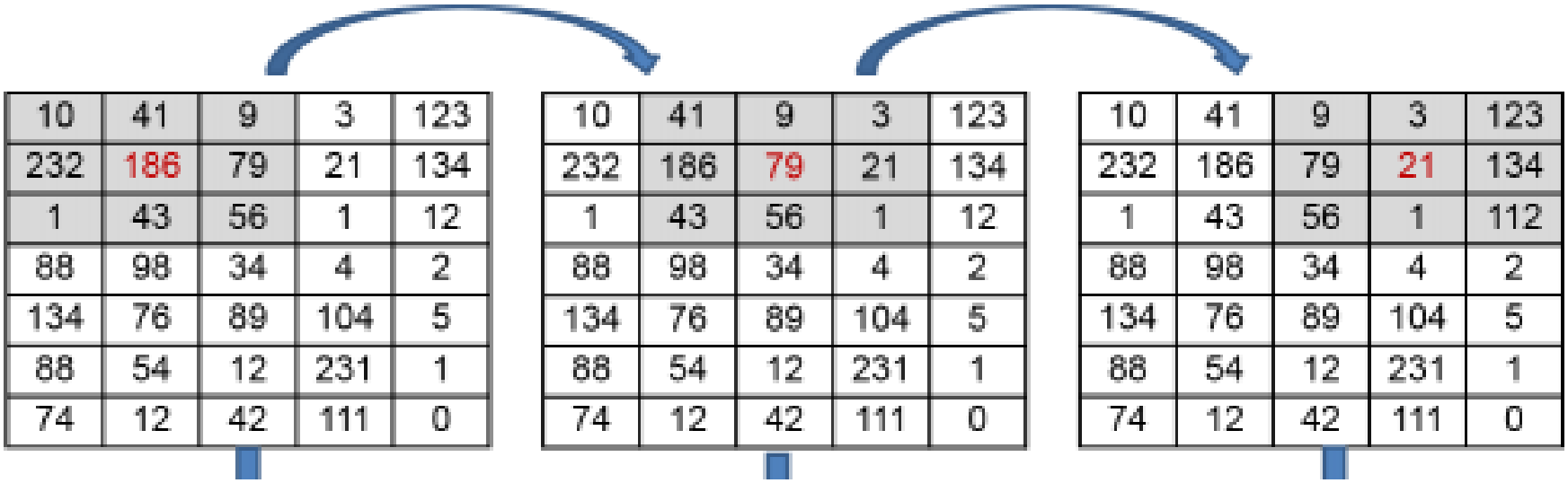
Image Net

https://image-net.org


ImageNet
https://image-net.org
IMAGENET
14,197,122 images, 21841 synsets indexed
Home Download Challenges About
Not logged in. Login | Signup
ImageNet is an image database organized according to the WordNet hierarchy (currently only the nouns), in which each node of the hierarchy is depicted by hundreds and thousands of images. The project has been instrumental in advancing computer vision and deep learning research. The data is available for free to researchers for non-commercial use.
Mar 11 2021. ImageNet website update.
© 2020 Stanford Vision Lab, Stanford University, Princeton University imagenet.help.desk@gmail.com Copyright infringement

Assignment-6

Enlist at least 5 image databases and their unique feature/distinction



10	41	9	3	123
232	186	79	21	134
1	43	56	1	12
88	98	34	4	2
134	76	89	104	5
88	54	12	231	1
74	12	42	111	0

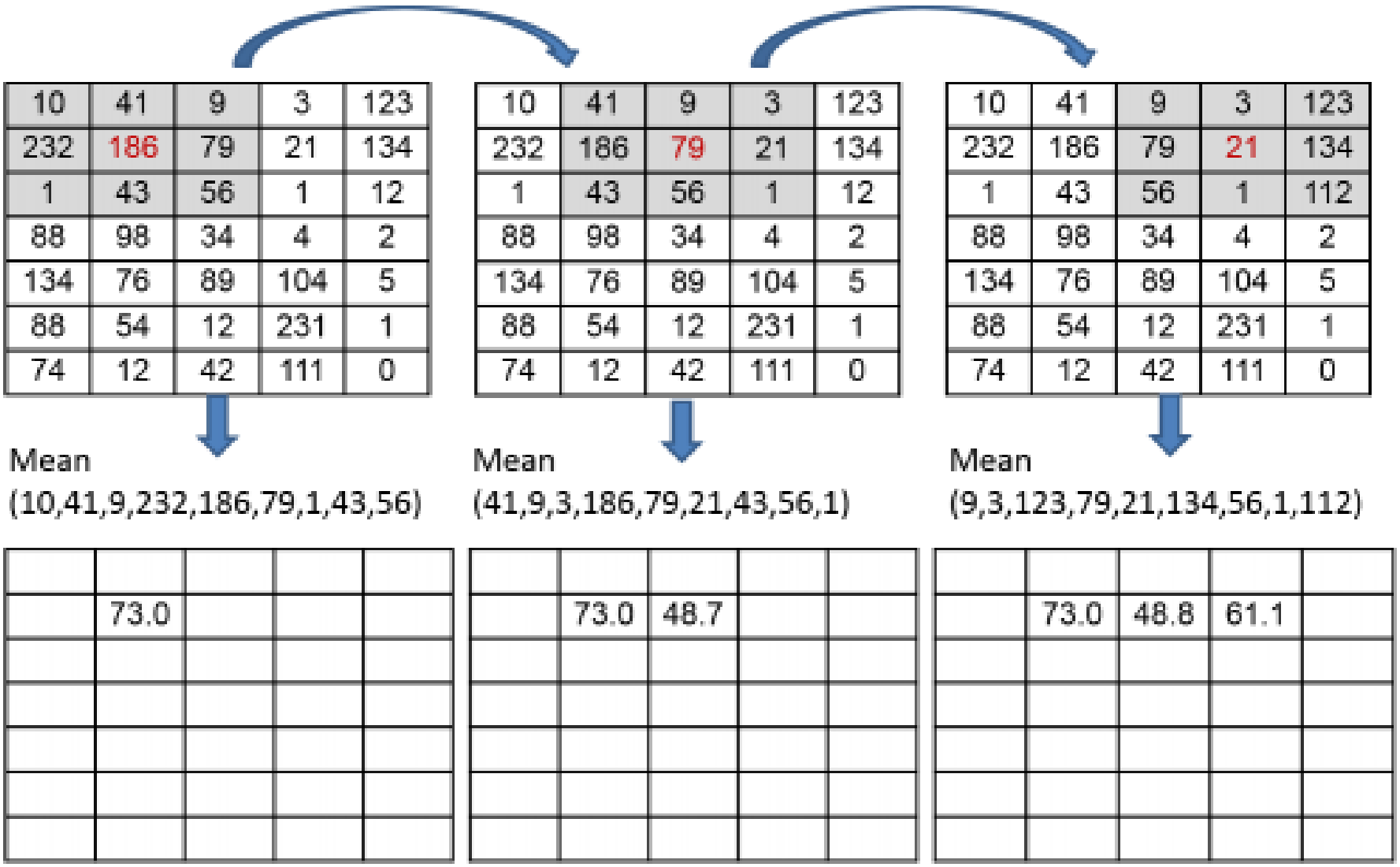
10	41	9	3	123
232	186	79	21	134
1	43	56	1	12
88	98	34	4	2
134	76	89	104	5
88	54	12	231	1
74	12	42	111	0

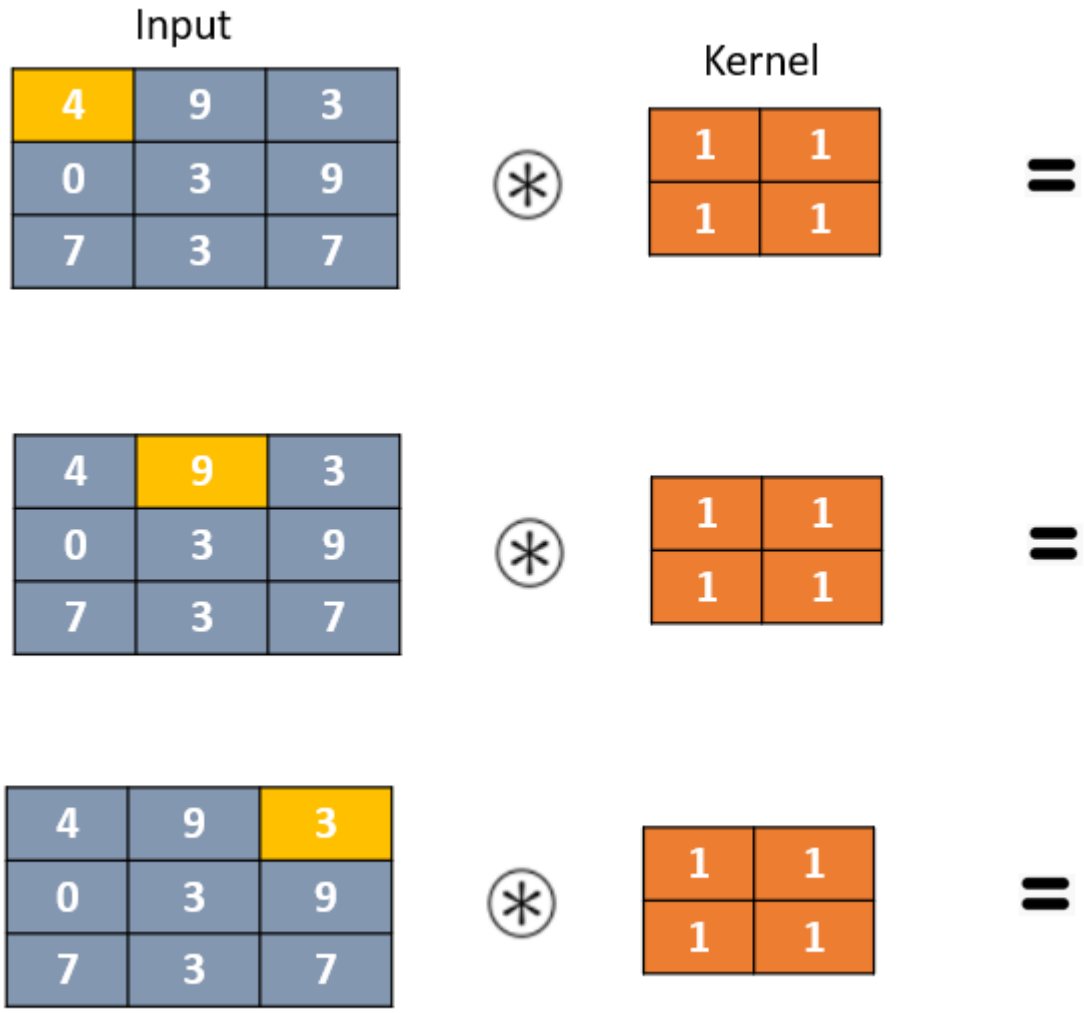
10	41	9	3	123
232	186	79	21	134
1	43	56	1	112
88	98	34	4	2
134	76	89	104	5
88	54	12	231	1
74	12	42	111	0

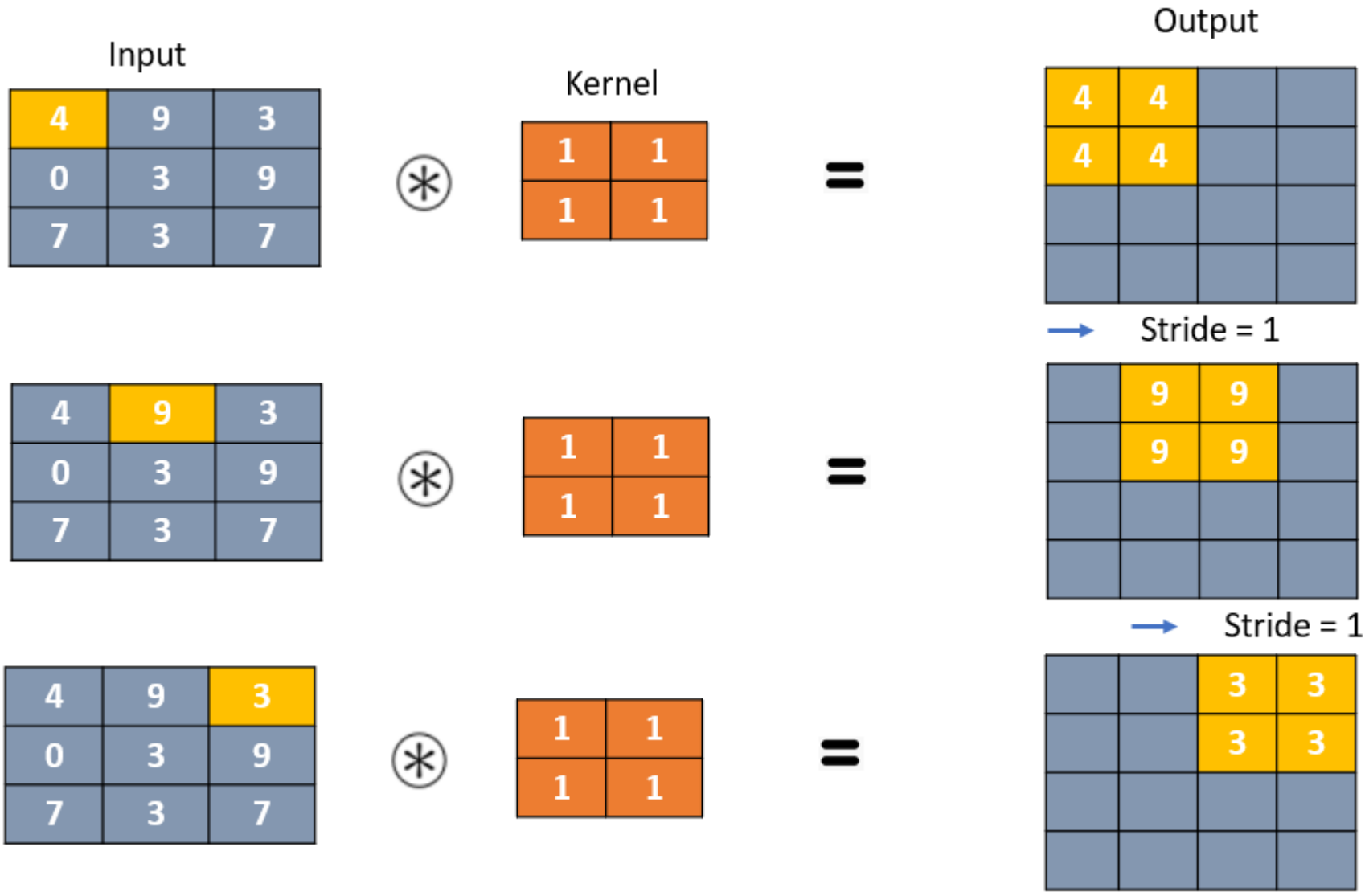
	73.0			

	73.0	48.7		

	73.0	48.8	61.1	

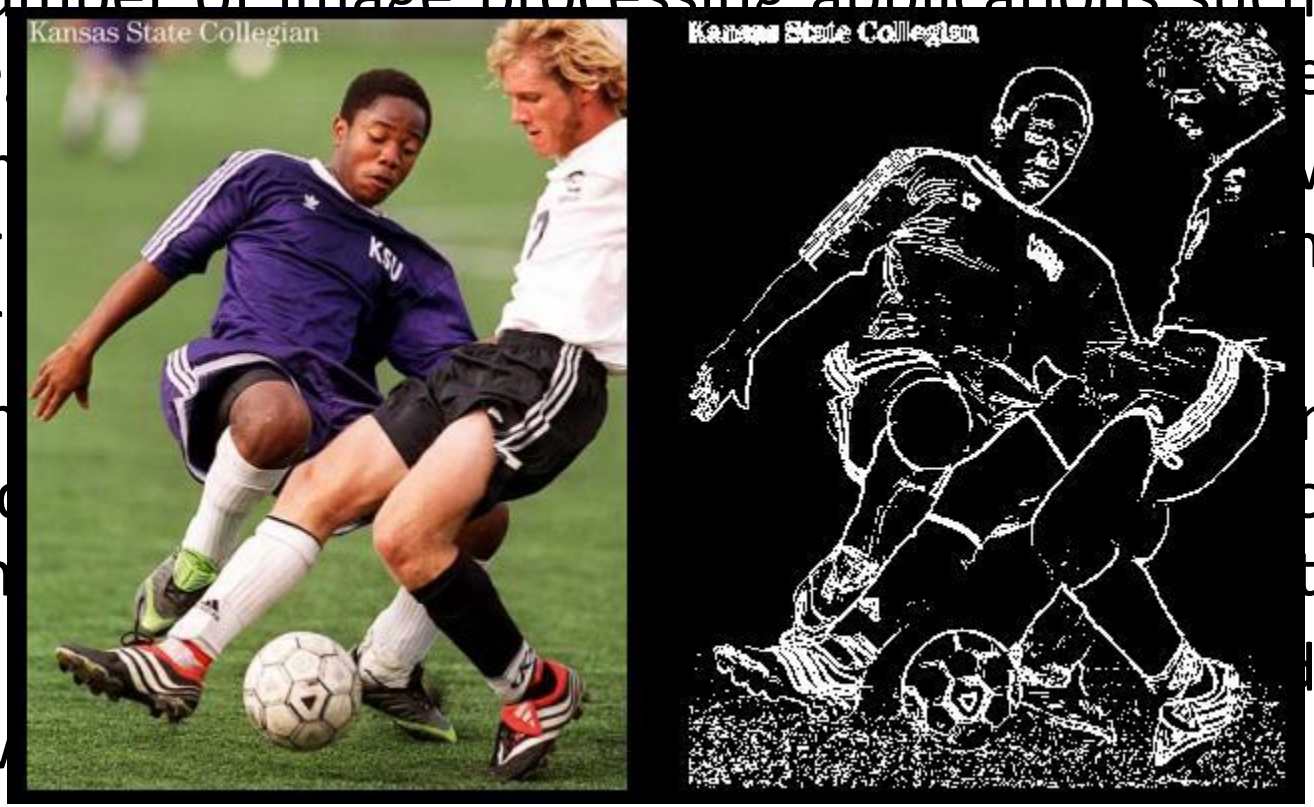

10 41 9 3 123
232 186 79 21 134
1 43 56 1 12
88 98 34 4 2
134 76 89 104 5
88 54 12 231 1
74 12 42 111 0
Mean
(10,41,9,232,186,79,1,43,56)
73.0
10 41 9 3 123
232 186 79 21 134
1 43 56 1 12
88 98 34 4 2
134 76 89 104 5
88 54 12 231 1
74 12 42 111 0
Mean
(41,9,3,186,79,21,43,56,1)
73.0 48.7
10 41 9 3 123
232 186 79 21 134
1 43 56 1 12
88 98 34 4 2
134 76 89 104 5
88 54 12 231 1
74 12 42 111 0
Mean
(9,3,123,79,21,134,56,1,112)
73.0 48.8 61.1


Input
Kernel
4 9 3
0 3 9
7 3 7
1 1
1 1
4 9 3
0 3 9
7 3 7
1 1
1 1
4 9 3
0 3 9
7 3 7
1 1
1 1

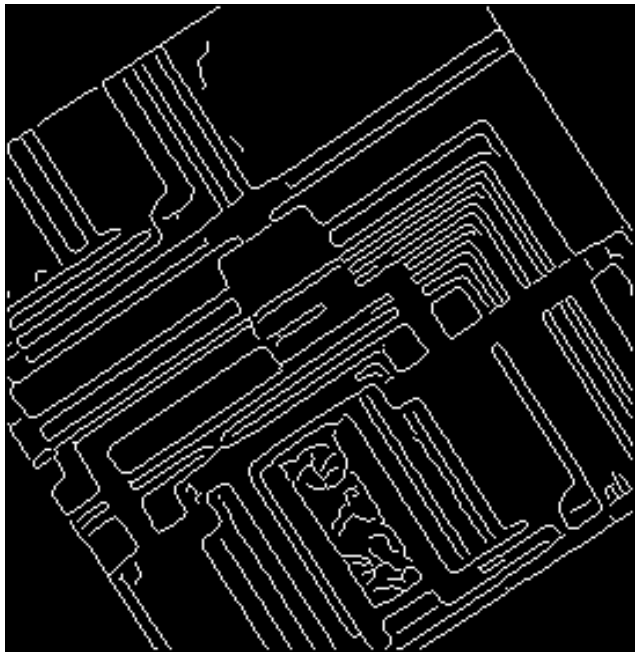

Input
Kernel
Output
4 9 3
0 3 9
7 3 7
1 1
1 1
4 4
4 4
Stride = 1
9 9
9 9
Stride = 1
3 3
3 3

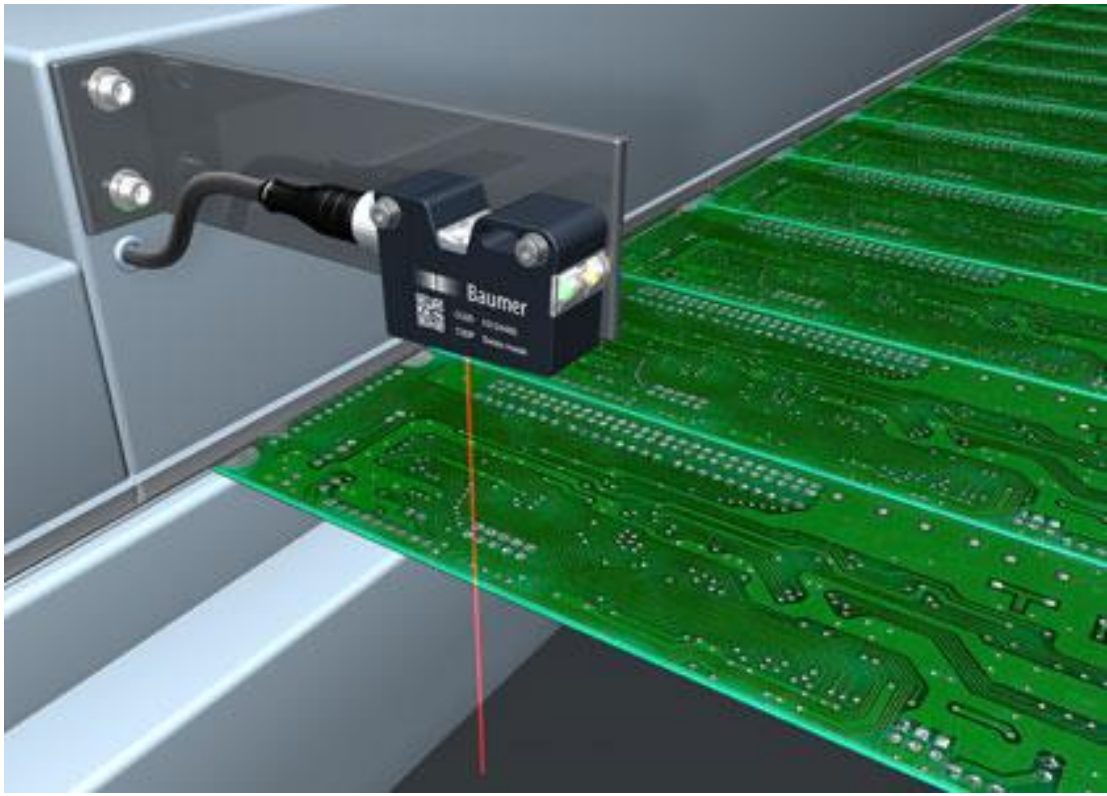
Edge & Line Detection

- Edge detection is an important operation in a large number of image processing applications such as image se e analysis.
- An which the br magnitude or
- An n intensity inc ons in an im ty
- A d between tw

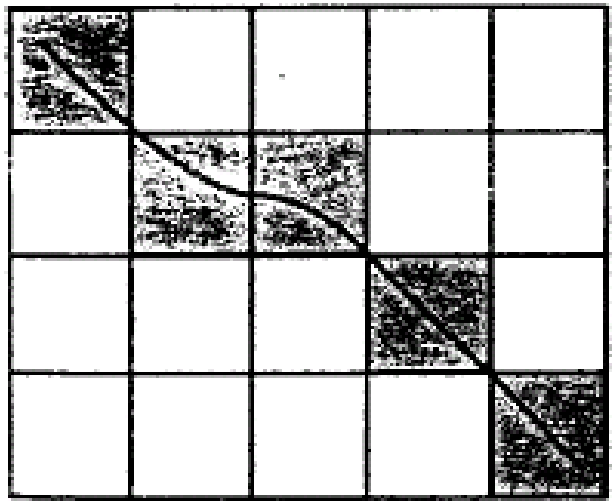


Edge & Line Detection

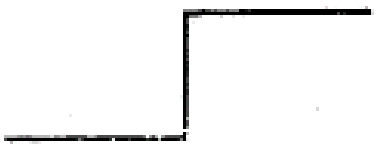




Edges & Lines



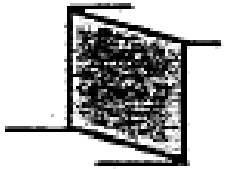
Connected edge points



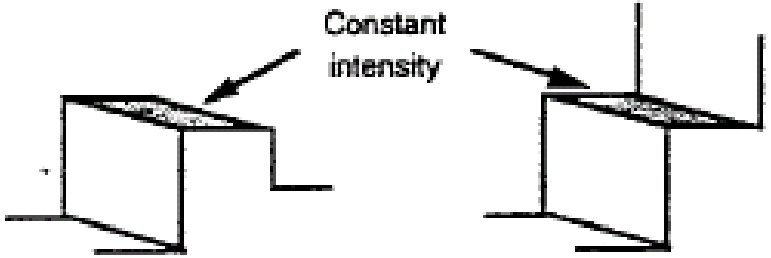
Ideal edge

Real edge

Noisy edge



Edge



Lines – a pair of edges which act as the boundary of a thin region of constant intensity

Edge Filters

Method 1

-1	-2	-1
0	0	0
+1	+2	+1

-2	-1	0
-1	0	+1
0	+1	+2

-1	0	+1
-2	0	+2
-1	0	+1

0	1	2
-1	0	1
-2	-1	0

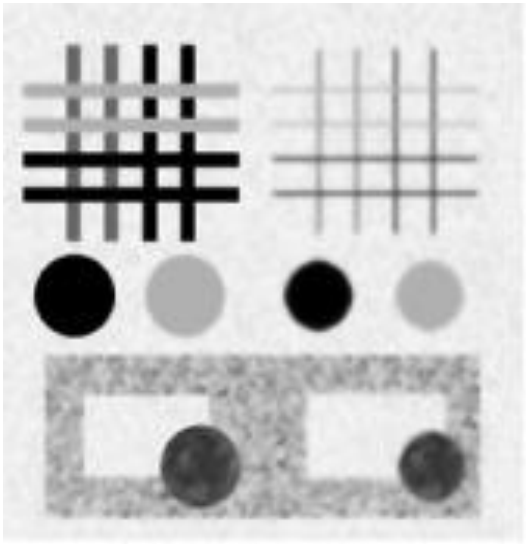
Method 2

-1	-1	-1
+2	+2	+2
-1	-1	-1

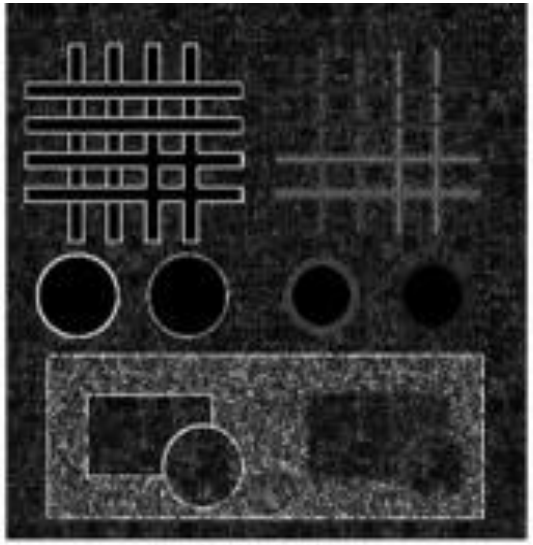
-1	-1	+2
-1	+2	-1
+2	-1	-1

-1	+2	-1
-1	+2	-1
-1	+2	-1

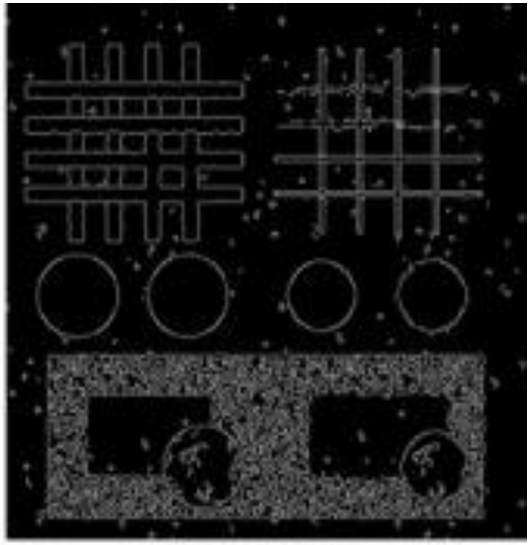
+2	-1	-1
-1	+2	-1
-1	-1	+2



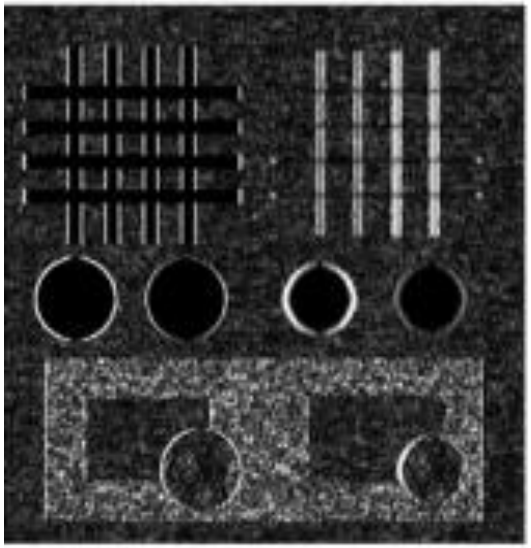
Original



Laplacian

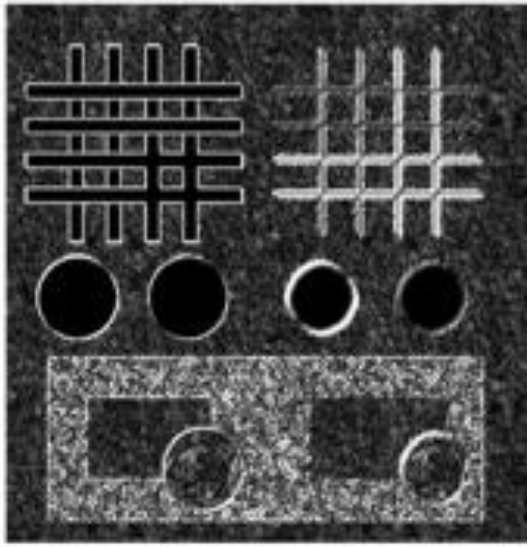


Canny



Sobel X

Sobel Y



Sobel X+Y

Edge Enhancement

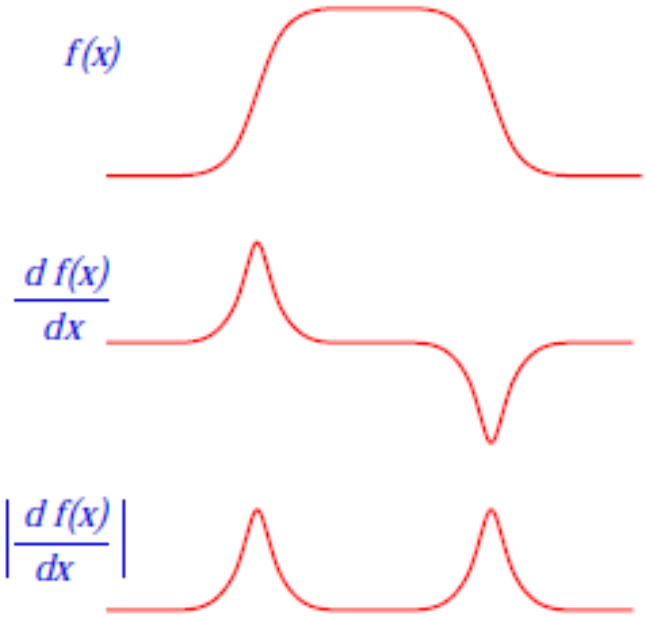
- Another class of filters can be defined by their effect of emphasizing or strengthening the edges within an image.
- In general a high pass filter has the inverse characteristic of a low pass filter; it will not change the high frequency component of the signal but will attenuate the low frequencies and eliminate any constant background intensity.
- High pass filter has the property that pixel data in the region of an object edge is modified and the edge effect is enhanced.
- This property of edge detection is a prerequisite for many image analysis operations.

Edge Enhancement

- There are four distinct approaches to the problem of edge detection:
  - Gradient and difference based operators
  - Template matching
  - Edge fitting
  - Statistical edge detection

Edge Detection

- It is the detection of change in intensity values, i.e. can be modeled as differential of image function



$$\frac{\partial f}{\partial x} = f(x+1) - f(x).$$

- We need to define a threshold

$$\left|\frac{\mathrm{d}\, f(x)}{\mathrm{d}x}\right| > T \quad \Rightarrow \text{Edge}$$

Edge Detection

$\left|\frac{\partial f(x,y)}{\partial x}\right|$ → Vertical Edges $\left|\frac{\partial f(x,y)}{\partial y}\right|$ → Horizontal Edges

$$H_1 = \begin{bmatrix} -1 & 0 & 1 \\ -1 & 0 & 1 \\ -1 & 0 & 1 \end{bmatrix} \text{ and } H_2 = \begin{bmatrix} -1 & -1 & -1 \\ 0 & 0 & 0 \\ 1 & 1 & 1 \end{bmatrix}.$$

-1	0	+1
-2	0	+2
-1	0	+1

x filter

+1	+2	+1
0	0	0
-1	-2	-1

y filter

Why Second Derivative?

- In most applications, the second derivative is better suited than the first derivative for image enhancement because of the ability of the former to enhance fine detail.
- Laplacian Operator

$$\nabla^2 f = \frac{\partial^2 f}{\partial x^2} + \frac{\partial^2 f}{\partial y^2}.$$

$$\frac{\partial^2 f}{\partial^2 x^2} = f(x+1, y) + f(x-1, y) - 2f(x, y)$$

$$\frac{\partial^2 f}{\partial^2 y^2} = f(x, y+1) + f(x, y-1) - 2f(x, y)$$

$$\nabla^2 f = [f(x+1, y) + f(x-1, y) + f(x, y+1) + f(x, y-1)] - 4f(x, y).$$

Implementation of Laplacian Operator

0	1	0
1	−4	1
0	1	0

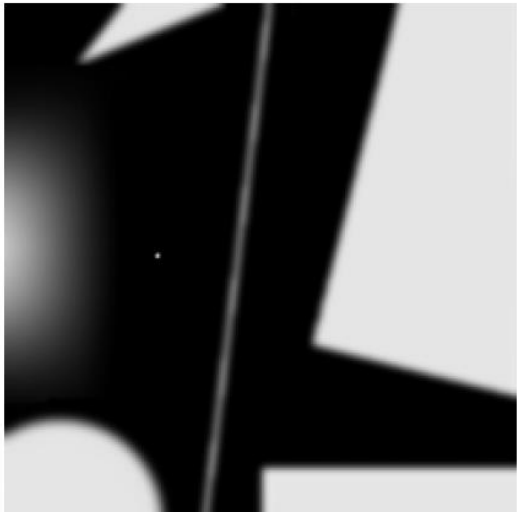
1	1	1
1	−8	1
1	1	1

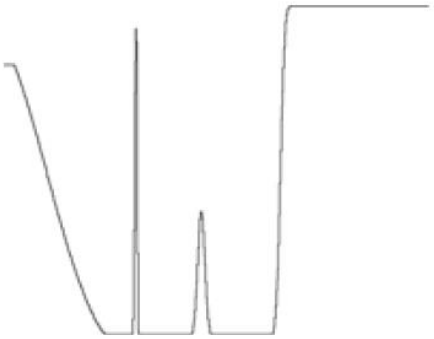
0	−1	0
−1	4	−1
0	−1	0

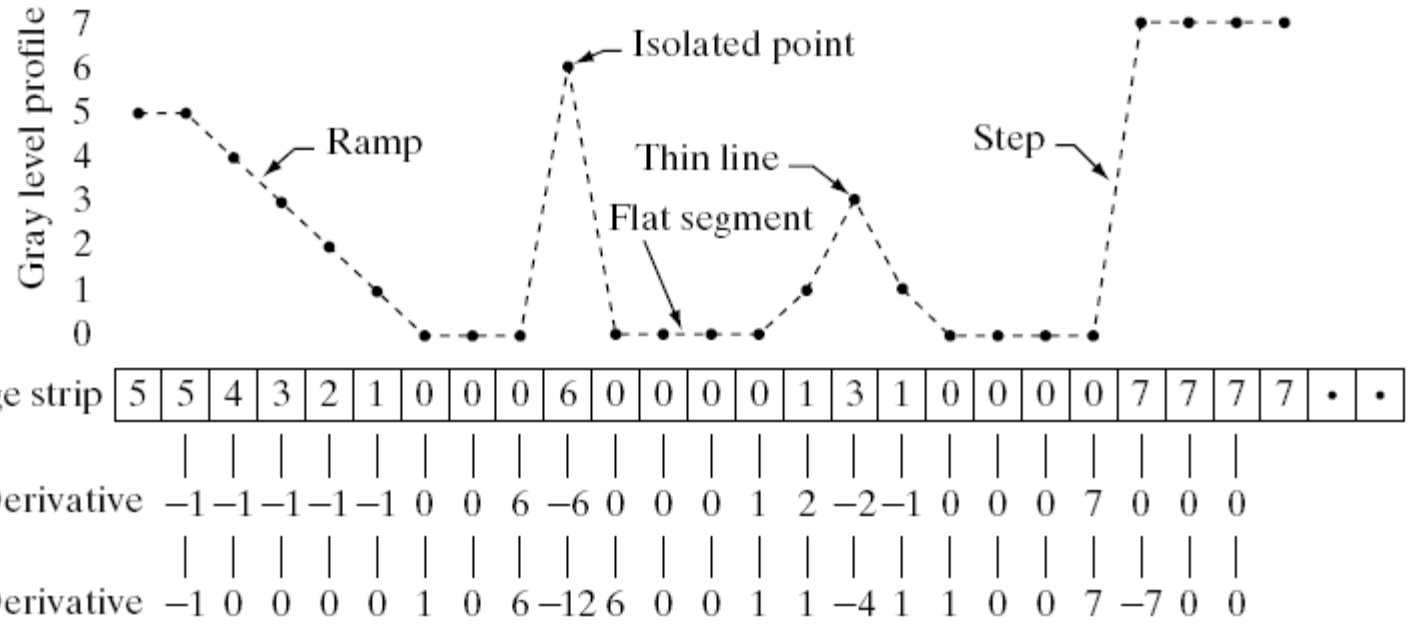
−1	−1	−1
−1	8	−1
−1	−1	−1

$$g(x, y) = \begin{cases} f(x, y) - \nabla^2 f(x, y) & \text{if the center coefficient of the Laplacian mask is negative} \\ f(x, y) + \nabla^2 f(x, y) & \text{if the center coefficient of the Laplacian mask is positive.} \end{cases}$$

Detecting the Change






Gray level profile
7
6
5
4
3
2
1
0
Ramp
Isolated point
Thin line
Flat segment
Step
Image strip
5 5 4 3 2 1 0 0 0 6 0 0 0 0 1 3 1 0 0 0 0 7 7 7 7
First Derivative
−1 −1 −1 −1 −1 0 0 6 −6 0 0 0 1 2 −2 −1 0 0 0 7 0 0 0
Second Derivative
−1 0 0 0 0 1 0 6 −12 6 0 0 1 1 −4 1 1 0 0 7 −7 0 0

Application of Laplacian

a b
c d

FIGURE 3.40
(a) Image of the North Pole of the moon.
(b) Laplacian-filtered image.
(c) Laplacian image scaled for display purposes.
(d) Image enhanced by using Eq. (3.7-5).
(Original image courtesy of NASA.)

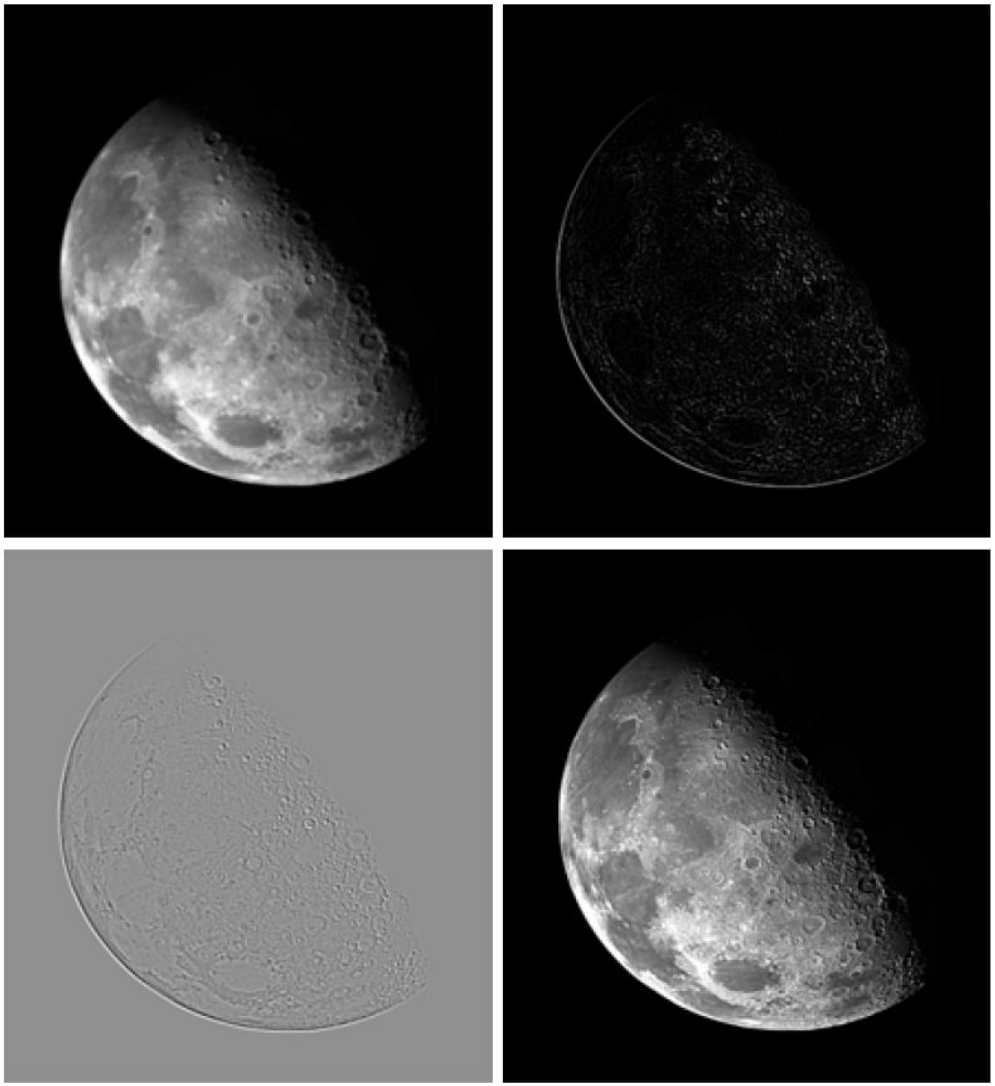


Image Sharpening

MATLAB Results

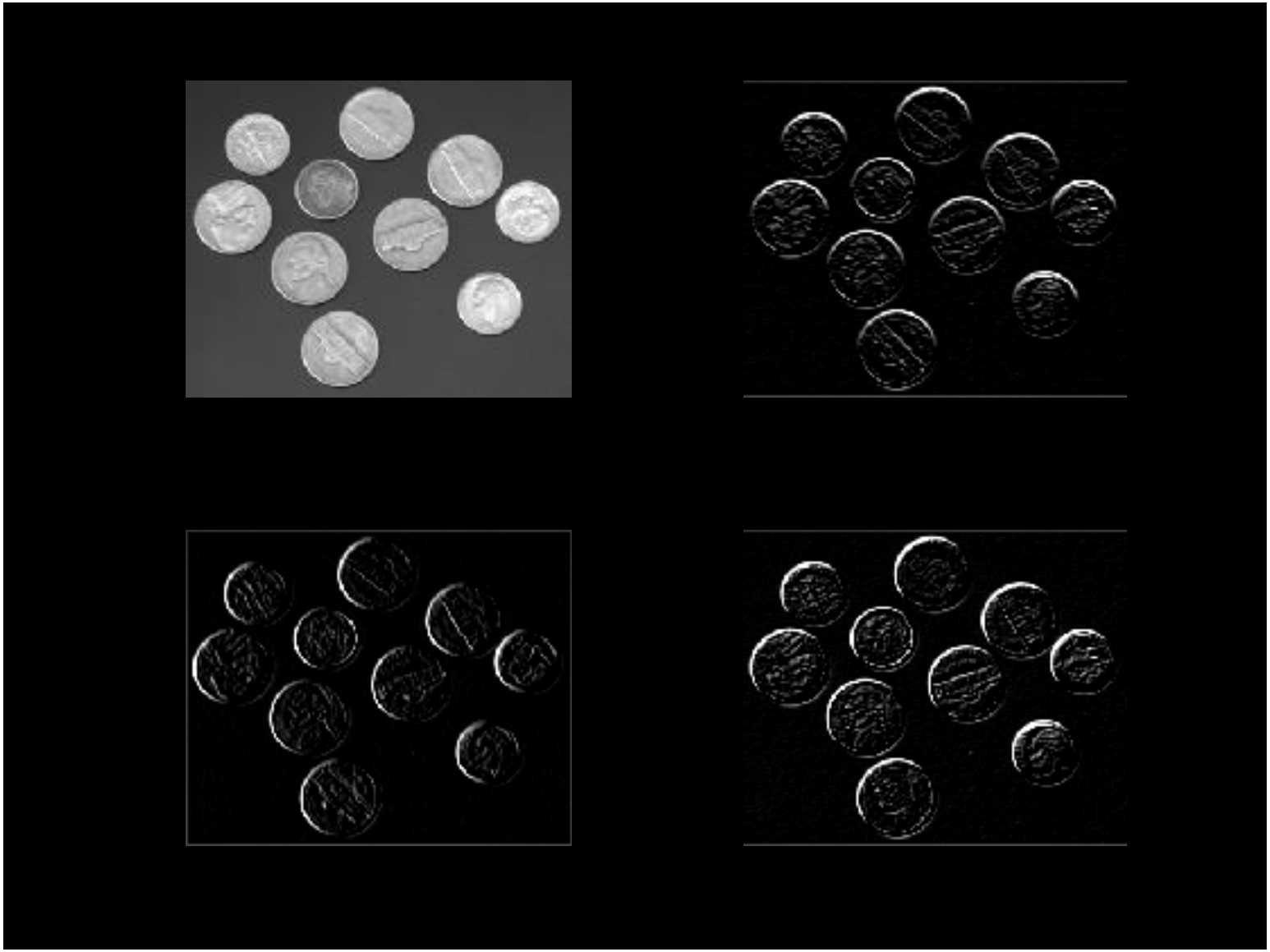
Original Image

Sobel Horizontal

Sobel Vertical

Sobel Combined

MATLAB Results



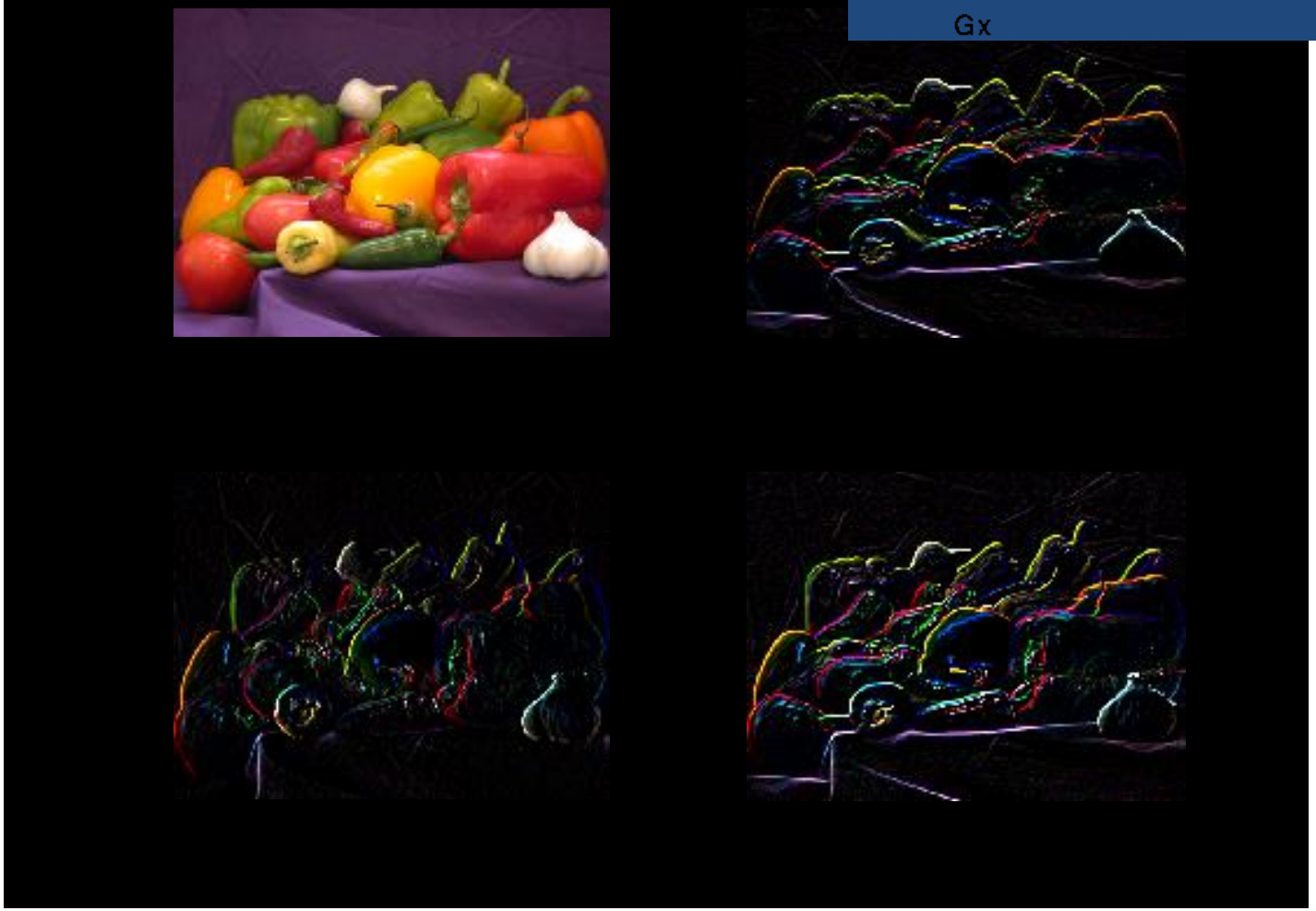
MATLAB Resu

-1	0	+1
-2	0	+2
-1	0	+1

Gx

+1	+2	+1
0	0	0
-1	-2	-1

Gy



MATLAB Results